

EXAMINER'S AMENDMENT

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

2. Authorization for this examiner's amendment was given in a telephone interview with Mr. Alan W. Young on August 7, 2008.

3. The applicant has been amended as follow:

In the claims:

1. (Currently amended) A method of servicing a request for a document over a computer network, comprising the steps of:

receiving a request for a document, the [[received]] requested document including a script that defines plurality of blocks, each block including a reference to a data source and code that is adapted to access the data source and to format the data accessed from the data source;

retrieving only some of the plurality of blocks defined in the script of the requested document from a memory, the memory storing the at least one of the plurality of blocks defined in the script of the requested document;

dynamically generating ~~[[remaining ones those]]~~ remaining blocks of the plurality of blocks defined in the script of the requested document that were not retrieved from the memory, that are not stored in the memory or that are stored in the memory but have been invalidated and storing a copy of each dynamically generated block in the memory;

assembling the requested document from both the retrieved and dynamically generated blocks, and

sending the assembled document over the computer network to an originator of the request.

2-3. (Canceled)

4. (Previously Presented) The method of claim 1, further comprising the step of sending at least one of the retrieved and dynamically generated blocks over the computer network to an originator of the request.

5. (Original) The method of claim 1, wherein the document includes an XML document.

6. (Original) The method of claim 5, wherein the document includes an HTML document.

7. (Original) The method of claim 1, wherein the request includes an HTTP request.

8. (Original) The method of claim 1, wherein the memory is a cache memory.

9. (Original) The method of claim 1, wherein the memory is adapted to be shared among multiple processes.

10. (Original) The method of claim 1, further including the step of determining

Art Unit: 2144

whether the at least one of the plurality of stored blocks has been invalidated and carrying out the retrieving step only when the at least one of the plurality of stored blocks has not been invalidated.

11. (Original) The method of claim 1, further comprising the step of determining at least one of an invalidation mechanism and an expiration time for each dynamically generated block that is stored in the memory.

12. (Original) The method of claim 1, further comprising the step of storing a placeholder block configured to enable an external data source to asynchronously publish data thereto.

13. (Original) The method of claim 12, wherein the placeholder block is free of code to access and format data.

14. (Original) The method of claim 1, further comprising the step of accepting asynchronous input from an external data source, the asynchronous input updating at least one block stored in the memory.

15. (Original) The method of claim 1, wherein the memory is maintained across a plurality of cache servers, and wherein a coherency mechanism maintains coherency of the memory across the plurality of cache servers.

16. (Original) The method of claim 15, wherein the plurality of cache servers are distributed over a geographical area.

17. (Original) The method of claim 1, further comprising the steps of associating at least one caching property to each dynamically generated block, the at least one caching property determining when the associated block is invalidated.

Art Unit: 2144

18. (Original) The method of claim 17, wherein the at least one caching property is stored along with the copy of each dynamically generated block stored in the memory.

19. (Currently Amended) A computer system for servicing a request for a document over a computer network, comprising:

at least one processor;

at least one data storage device;

a plurality of processes spawned by said at least one processor, the processes including processing logic for:

receiving a request for a document, the [[received]] requested document including a script that defines plurality of blocks, each block including a reference to a data source and code that is adapted to access the data source and to format the data accessed from the data source;

retrieving only some of the plurality of blocks defined in the script of the requested document from a memory, the memory storing the at least one of the plurality of blocks defined in the script of the requested document;

dynamically generating ~~[[remaining ones~~ those]]

remaining blocks of the plurality of blocks defined in the script of the requested document that were not retrieved from the memory, that are not stored in the memory or that are stored in the memory but have been invalidated and storing a copy of each dynamically generated block in the memory;

assembling the requested document from both the retrieved and dynamically generated blocks, and

sending the assembled document over the computer network to an originator of the request.

20-21. (Canceled).

22. (Previously Presented) The computer system of claim 19, further comprising processing logic for sending at least one of the retrieved and dynamically generated blocks over the computer network to an originator of the request.

23. (Original) The computer system of claim 19, wherein the document includes an XML document.

24. (Original) The computer system of claim 23, wherein the document includes an HTML document.

25. (Original) The computer system of claim 19, wherein the request includes an HTTP request.

26. (Original) The computer system of claim 19, wherein the memory is a cache memory.

27. (Original) The computer system of claim 19, wherein the memory is adapted to be shared among multiple processes.

28. (Original) The computer system of claim 19, further including processing logic for determining whether the at least one of the plurality of stored blocks has been invalidated and for carrying out the retrieving step only when the at least one of the plurality of stored blocks has not been invalidated.

29. (Original) The computer system of claim 19, further comprising processing logic

for determining at least one of an invalidation mechanism and an expiration time for each dynamically generated block that is stored in the memory.

30. (Original) The computer system of claim 19, further comprising processing logic for storing a placeholder block configured to enable an external data source to asynchronously publish data thereto.

31. (Original) The computer system of claim 30, wherein the placeholder block is free of code to access and format data.

32. (Original) The computer system of claim 19, further comprising processing logic for accepting asynchronous input from an external data source, the asynchronous input updating at least one block stored in the memory.

33. (Original) The computer system of claim 19, wherein the memory is maintained across a plurality of cache servers, and wherein a coherency mechanism maintains coherency of the memory across the plurality of cache servers.

34. (Original) The computer system of claim 33, wherein the plurality of cache servers are distributed over a geographical area.

35. (Original) The computer system of claim 19, further comprising processing logic for associating at least one caching property to each dynamically generated block, the at least one caching property determining when the associated block is invalidated.

36. (Original) The computer system of claim 35, wherein the at least one caching property is stored along with the copy of each dynamically generated block stored in the memory.

37. (Currently amended) A machine-readable storage medium having data stored thereon representing sequences of instructions which, when executed by computing

Art Unit: 2144

device, causes said computing device to service a request for a document over a computer network, by performing the steps of:

receiving a request for a document, the [[received]] requested document including a script that defines plurality of blocks, each block including a reference to a data source and code that is adapted to access the data source and to format the data accessed from the data source;

retrieving only some of the plurality of blocks defined in the script of the requested document from a memory, the memory storing the at least one of the plurality of blocks defined in the script of the requested document;

dynamically generating ~~[[remaining ones those]]~~ remaining blocks of the plurality of blocks defined in the script of the requested document that were not retrieved from the memory, that are not stored in the memory or that are stored in the memory but have been invalidated and storing a copy of each dynamically generated block in the memory;

assembling the requested document from both the retrieved and dynamically generated blocks, and

sending the assembled document over the computer network to an originator of the request.

38-39. (Canceled).

40. (Previously Presented) The medium of claim 37, further comprising the step of sending at least one of the retrieved and dynamically generated blocks over the computer network to an originator of the request.

Art Unit: 2144

41. (Original) The medium of claim 37, wherein the document includes an XML document.
42. (Original) The medium of claim 41, wherein the document includes an HTML document.
43. (Original) The medium of claim 37, wherein the request includes an HTTP request.
44. (Original) The medium of claim 37, wherein the memory is a cache memory.
45. (Original) The medium of claim 37, wherein the memory is adapted to be shared among multiple processes.
46. (Original) The medium of claim 37, further including the step of determining whether the at least one of the plurality of stored blocks has been invalidated and carrying out the retrieving step only when the at least one of the plurality of stored blocks has not been invalidated.
47. (Original) The medium of claim 37, further comprising the step of determining at least one of an invalidation mechanism and an expiration time for each dynamically generated block that is stored in the memory.
48. (Original) The medium of claim 37, further comprising the step of storing a placeholder block configured to enable an external data source to asynchronously publish data thereto.
49. (Original) The medium of claim 48, wherein the placeholder block is free of code to access and format data.
50. (Original) The medium of claim 37, further comprising the step of accepting asynchronous input from an external data source, the asynchronous input updating at

least one block stored in the memory.

51. (Original) The medium of claim 37, wherein the memory is maintained across a plurality of cache servers, and wherein a coherency mechanism maintains coherency of the memory across the plurality of cache servers.

52. (Original) The medium of claim 51, wherein the plurality of cache servers are distributed over a geographical area.

53. (Original) The medium of claim 37, further comprising the steps of associating at least one caching property to each dynamically generated block, the at least one caching property determining when the associated block is invalidated.

54. (Original) The medium of claim 53, wherein the at least one caching property is stored along with the copy of each dynamically generated block stored in the memory.

55. (Currently Amended) A method of servicing a request for a Web page over a computer network, comprising the steps of:

identifying constituent blocks of the Web page, each of the constituent blocks including a portion of the Web page;

defining a caching property for each identified block, the caching property defining when each identified block is to be invalidated;

caching the identified blocks in a memory;

maintaining each of the cached blocks in the memory according to the defined caching property defined for each block;

servicing the request for the Web page by retrieving only some of the cached blocks of the requested Web page from the memory and ~~generating all remaining blocks~~

Art Unit: 2144

~~of the requested Web page that were not retrieved from the memory~~ dynamically generating [[those]] remaining blocks of the constituent blocks of the requested Web page that were not retrieved from the memory, that are not stored in the memory or that are stored in the memory but have been invalidated and storing a copy of each dynamically generated block in the memory;

assembling the Web page from both the retrieved and dynamically generated blocks, and

sending the assembled Web page over the computer network.

56. (Original) The method of claim 55, wherein each of the constituent blocks includes a reference to a data source and code that is adapted to access the data source and to format the data accessed from the data source.

57. (Canceled).

58. (Previously Presented) The method of claim 55, further including a step of storing a copy of any generated block in the memory.

59. (Canceled).

60. (Original) The method of claim 55, wherein the caching properties include at least one of a unique identifier, an expiration date, an expiration time and an invalidation rule.

4. The following is an examiner's statement of reasons for allowance:

With respect to claims 1, 4-19, 22-37, 40-56, 58, and 60, the prior art of record, individually or in combination, fails to teach, suggest or render obvious the claimed

Art Unit: 2144

invention in combination with the specific amended limitations as recited in claims 1, 19, 37, and 55. Also see Applicant's argument remark filed on May 9, 2008.

5. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

6. The Petition 1.48-petition to correct inventorship filed on April 8, 2003, is being considered by the Examiner

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tammy T. Nguyen whose telephone number is 571-272- 3929. The examiner can normally be reached on Monday - Friday 8:30 - 5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, ***William Vaughn*** can be reached on 571-272-3922. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business

Art Unit: 2144

Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Thanh Tammy Nguyen/

Thanh Tammy Nguyen

Primary Examiner, Art Unit 2144